Computing Nash Equilibria of Unbounded Games

Martín Escardó[1] and Paulo Oliva[2]*

[1] University of Birmingham
School of Computer Science
m.escardo@cs.bham.ac.uk
[2] Queen Mary University of London
School of Electronic Engineering and Computer Science
paulo.oliva@eecs.qmul.ac.uk

Abstract

Using techniques from higher-type computability theory and proof theory we extend the well-known game-theoretic technique of *backward induction* to certain general classes of unbounded games. The main application is a *closed formula* for calculating strategy profiles in Nash equilibrium and subgame perfect equilibrium even in the case of games where the length of play is not a-priori fixed.

1 Introduction

In his short 1912 article Zermelo [25] asked[1] the following interesting question: What properties does a chess position q has to have so that White, independently of how Black plays, can *force* a win in at most r moves? Zermelo goes on to give a complete characterisation of these positions based on the non-emptiness of the union of a certain family of sets $U_r(q)$ determined by the position q and the bound r. Early work such as this of Zermelo (amongst many others) led von Neumann in 1928 [24] to ask an even more interesting questions: How should each player actually choose their moves to achieve an optimal outcome in a game? So, rather than just trying to identify what positions are "check-mate in r-moves", one is actually interested in calculating how to guarantee reaching such positions whenever this is possible. This was the start of a mathematical theory of games which culminated with von Neumann and Morgenstern's seminal book [23] providing the foundations for a new branch of Mathematics and Economics today known as Game Theory.

It was in this very first book [23] on Game Theory that the technique of *backward induction* as a way of calculating a player's "optimal strategy" first appears (cf. Schwalbe [20]). Informally, backward induction is a way of calculating optimal strategies for each round of the game by starting with the last round. In the last round of the game it is easy to check what move will lead to the best outcome. The idea is then to fix this as *the* strategy for the last round. One can then go one step back and calculate the optimal strategy for the last-but-one round: For each possible move check what the last player will do according to his optimal strategy (which we just fixed) and see what outcome that will lead to. The last-but-one player should pick a move which leads to the best outcome amongst the possible outcomes. Proceeding like this one can determine the best way each player can play at each round[2]. In modern terminology, one

*Supported by the Royal Society grant 516002.K501/RH/kk

[1]For an English translation of Zermelo's 1912 article see [20].

[2]For more information on backward induction see Section 2, or [2], [14] (section 4.2), [17] (section 7.7) and [18] (chapter 3).

actually says that such set of strategies is in *subgame perfect equilibrium*, a stronger notion that than of Nash equilibrium (see Section 2).

The technique of backward induction, however, has so far only been considered for games where the number of players is finite and fixed a priori, and more importantly, that the game tree itself is finite. For instance, Aliprantis [1] stresses that

"*The finiteness of the [game] tree as well as the complete information hypothesis are the two essential ingredients for the [backward induction] method to work.*"

As discussed in [12], chess itself is a potentially infinite game, so that the technique of backward induction does not apply in the strict sense. In the case of chess, however, there exists a strategically equivalent variant of chess which is finite (cf. [12]). Because the number of move choices is finite at each point, the game tree has a fixed depth. But what about games where the number of moves at each round might be infinite, so that the game tree itself will be infinite (even if the length of plays is finite and fixed). A simple example of such a game is the Sweet Tooth game [21] where the first player can cut the two given cakes into two pieces with arbitrary precision (see Section 2.3).

Our main contribution in this paper is a generalisation of backward induction method for *well-founded games*, i.e. games which always terminate after a finite number of rounds, but whose length of plays can be arbitrarily long depending on how the game is played – so length of plays are not a priori bounded. We also allow for infinite sets of moves, so that the game tree can possibly be infinitely branching as well.

Our definition of the backward induction method is completely formal, and relies on some extensions of the simply typed λ-calculus. This is in stark contrast with the current uses of backward induction in the literature, which allow for confusion when comparing different applications of the method to slightly different games. For instance, Boves [7] writes that "there seems to be a similarity between the backward induction argument for the finite iterated prisoner's dillema and the surprise exam paradox and one cannot help but wonder whether the former is indeed no more than an instance of the latter". Iin here we give a precise *mathematical description* of the backward induction method, together with a closed formula for the strategy profile in sub-game perfect equilibrium. A formal proof that the backward induction method leads to a strategy profile in subgame perfect equilibrium has recently also been given by Aliprantis [1] for *finite* games in extensive form. In our approach we consider games in normal form, but crucially, extend Aliprantis results in two ways:

- We consider games whose game tree have *infinitely many nodes*, but are nevertheless well-founded so that each play in the game eventually leads to an outcome.
- We provide a *closed formula* describing the resulting optimal strategy profile.

The work presented makes fundamental use of techniques from proof theory, higher-order computability theory, and our recent joint work [9, 10, 11] on selection functions and their products (see Section 2.2).

1.1 Higher-type functions and assumptions

We work with functions defined on sets of functions, and we denote the set of all functions from a set X to a set Y by $(X \to Y)$. A trivial example is the evaluation functional $E\colon (\mathbb{R} \to \mathbb{R}) \to \mathbb{R}$ defined by $E(f) = f(0)$. Sometimes we use Church's lambda notation to define such functions, where e.g. E is written simply $\lambda f.f(0)$, which is equivalent to the notation $f \mapsto f(0)$.

A typical example of a higher-type function in game theory is the functional

$$\mathsf{argsup}\colon (X \to R) \to X,$$

where X is any set and $R \subseteq \mathbb{R}$ is a set of real numbers. The functional argsup takes a function $f\colon X \to R$ as an argument, and returns *any point* where f attains its maximum value. That is,

$$f(\mathsf{argsup}(f)) \geq f(x), \text{ for all } x.$$

This functional is not well defined in general, but it is if X or R are finite, or if X is compact and $R = \mathbb{R}$ and we restrict attention to the continuous functions $X \to R$. In the body of the paper we assume that such a condition is satisfied, as we make crucial uses of argsup in our formalisation of backward induction.

As we shall see, argsup is one example of a *selection function*, among many others that also are relevant for game theory (cf. Section 2.2). Our main construction involves a higher nesting of function sets than argsup: it takes two or more selection functions as arguments and has another selection function as its value. When the selection functions are argsup, this procedure computes subgame perfect equilibria, as we shall see.

2 Sequential Games and Equilibrium

Before we proceed to explain our generalisation of backward induction to unbounded games, let us first see how we can give a completely formal description of backward induction for finite games. Let $\mathbf{n}$ denote the set $\{1, 2, \ldots, n\}$, which we think of as identifying n players.

Definition 2.1 (Finite sequential game)**.** *An n-player sequential game is given by a tuple $(\{X_i\}_{i \in \mathbf{n}}, q)$ where*

- *X_i is the set of moves for player i, and*
- *$q\colon X_1 \times \ldots \times X_n \to \mathbb{R}^n$ is the outcome function.*

A play in the game is a tuple $\vec{x} = x_1, \ldots, x_n$ in $X_1 \times \ldots \times X_n$. For each play $\vec{x}$ the n-tuple of real number $q(\vec{x})$ describes the payoffs obtained by each of the n players in that run of the game.

We remark that considering n distinct players, one for each round, is more general than the case of two-player games. The case of two players can be easily modelled via an outcome function which always produces n-tuples of payoffs $\vec{v}\colon \mathbb{R}^n$ where $v_i = v_{i+2}$, so that the set of players at even rounds and the set of players at odd rounds can be viewed as single players.

Definition 2.2 (Strategies)**.** *A* strategy *for player i is a mapping*

$$\mathsf{s}_i\colon X_1 \times \ldots \times X_{i-1} \to X_i$$

describing what move player i should chose based on what has been played up to point $i-1$. A strategy profile *is a tuple of strategies $(\mathsf{s}_i)_{i \in \mathbf{n}}$ for each player.*

An strategy profile $(\mathsf{s}_i)_{i \in \mathbf{n}}$ determines a play $\vec{x}$ as

$$x_i = \mathsf{s}_i(x_0, x_1, \ldots, x_{i-1})$$

and its corresponding outcome $u = q(\vec{x})$. We write $q_i\colon X_1 \times \ldots \times X_n \to \mathbb{R}$ for the composition of q with the i-projection. Hence, given a play $\vec{x}$, the payoff of player i is given by $q_i(\vec{x})$. Slightly abusing notation, we will also write $q_i(\mathsf{s}_1, \mathsf{s}_2, \ldots, \mathsf{s}_n)$ for the i-coordinate of the outcome determined by the strategy profile $(\mathsf{s}_i)_{i \in \mathbf{n}}$.

Definition 2.3 (Nash equilibrium)**.** *A strategy profile $(\mathsf{s}_i)_{i\in\mathbf{n}}$ is in Nash equilibrium if for each player i and alternative strategy s_i^* we have*

$$q_i(\mathsf{s}_1,\dots,\mathsf{s}_i,\dots,\mathsf{s}_n) \geq q_i(\mathsf{s}_1,\dots,\mathsf{s}_i^*,\dots,\mathsf{s}_n).$$

Informally, a strategy profile is in equilibrium if no player has an incentive to unilaterally change his strategy.

Definition 2.4 (Subgame)**.** *In any n-player sequential game $(\{X_i\}_{i\in\mathbf{n}}, q)$ a partial play $x_1,\dots,x_i$ determines a $(n-i)$-player subgame $(\{X_k\}_{i<k\leq n}, \tilde{q})$ with payoff function*

$$\tilde{q}_j(x_{i+1},\dots,x_n) = q_j(x_1,\dots,x_i,x_{i+1},\dots,x_n)$$

for $i<j\leq n$.

Definition 2.5 (Subgame Perfect Equilibrium)**.** *A strategy profile $(\mathsf{s}_i)_{i\in\mathbf{n}}$ is in* subgame perfect equilibrium *if it is in Nash equilibrium on any subgame.*

A strategy profile is in subgame perfect equilibrium if at each point in the game $x_1,\dots,x_{i-1}$ the current player i will not get a better payoff by making a different move y_i which deviates from his current choice of move $x_i = \mathsf{s}_i(x_1,\dots,x_{i-1})$. Note that for a Nash equilibrium this only needs to happen along the actual play determined by the strategy. A classical example of a game which has more Nash equilibria than subgame perfect equilibria is the ultimatum game [13].

2.1 Backward induction (informally)

Let $\mathsf{argsup}\colon (X_n \to \mathbb{R}) \to X_n$ denote the functional which returns any point $x\in X$ on which its argument function $p\colon X\to\mathbb{R}$ has maximum value. Given a finite sequential game as described in Definition 2.1, a strategy profile in subgame perfect equilibrium can be calculated as follows. First define the strategy for the last player n, which should be as follows:

$$\mathsf{s}_n(x_1,\dots,x_{n-1}) = \mathsf{argsup}(\lambda x_n.q_n(x_1,\dots,x_{n-1},x_n)).$$

That is, the last player's optimal strategy is simply to pick any move which gives himself maximum payoff. Recall that by $q_n(x_1,\dots,x_{n-1},x_n)$ we denote the payoff of player n. We can then proceed backwards to the last-but-one player $n-1$. As we have already fixed the optimal strategy of player n, the optimal strategy for player $n-1$ can be described as

$$\mathsf{s}_{n-1}(x_1,\dots,x_{n-2}) = \mathsf{argsup}(\lambda x_{n-1}.q_{n-1}(x_1,\dots,x_{n-2},x_{n-1},a_n(x_{n-1}))),$$

where the function

$$a_n(x_{n-1}) = \mathsf{s}_n(x_1,\dots,x_{n-2},x_{n-1})$$

calculates what move player n will choose for each different choice x_{n-1} of player $n-1$ move. We have then fixed the optimal strategies s_{n-1} and s_n of the last two players, and can then backtrack to compute the optimal strategy of player $n-2$ as

$$\mathsf{s}_{n-2}(x_1,\dots,x_{n-3}) = \mathsf{argsup}(\lambda x_{n-2}.q_{n-2}(x_1,\dots,x_{n-2},a_{n-1,n}(x_{n-2}))),$$

where the function

$$a_{n-1,n}(x_{n-2}) = \underbrace{\mathsf{s}_{n-1}(x_1,\dots,x_{n-2})}_{b_{n-1}}, \mathsf{s}_n(x_1,\dots,x_{n-2},b_{n-1},a_n(b_{n-1}))$$

calculates which moves player $n-1$ and player n will chose for each different choice of move by player $n-2$. Continuing like this one finally arrives at an optimal strategy for player one, completing the construction of a strategy profile in equilibrium.

This procedure for calculating optimal strategies is known as *backward induction.* Although it is easy to describe the first two or three steps, the calculations become unmanageable already for n bigger than 3. In the next section we show how backward induction can be easily formalised with the help of a binary operator on functionals such as argsup known as the product of selection functions.

2.2 Backward Induction Formalized

Following [9] we call a functional of type $(X \to R) \to X$ a selection function. Intuitively, X is the set of choices, and R is the set of outcomes. The mapping $X \to R$ describes what outcome results from each choice. Let us call such mapping $X \to R$ as *local outcome functions.* A selection function associates a particular move $x \in X$ for each local outcome function $p\colon X \to R$. Selection function on two sets of moves X and Y can be put together to build a selection function over the combined set $X \times Y$ as follows.

Definition 2.6 (Product of Selection Functions)**.** *Given two selection functions $\varepsilon\colon (X \to R) \to X$ and $\delta\colon (Y \to R) \to Y$, define their product $\varepsilon \otimes \delta$ which has type $((X \times Y) \to R) \to (X \times Y)$ using λ-notation as*

$$(\varepsilon \otimes \delta)(q^{X \times Y \to R}) = (a, y_a),$$

where $a = \varepsilon(\lambda x.q(x, y_x))$ and $y_x = \delta(\lambda y.q(x, y))$.

Consider a two-player game where the first player chooses a move in a set X followed by the second player choosing a move in the set Y. Hence, the product $X \times Y$ consists of the set of all possible plays of the game, R is to be viewed as the set of possible outcomes, and $q\colon X \times Y \to R$ the "outcome function" mapping plays to outcomes. Finally, the selection function ε and δ describe the preferred moves of each player when given a local outcome function $p\colon X \to R$ and $Y \to R$, respectively. The mapping y_x can be viewed as calculating for each move $x \in X$ of the first player, what move $y \in Y$ the second player will prefer to choose (according to δ). That's a strategy for the second player! The move a is then the preferred move (or strategy!) of the first player taking assuming the second (and last) player will follow strategy y_x. Therefore, the product of selection function completely captures the backward induction reasoning for simple games with two rounds.

Definition 2.7 (Iterated Product of Selection Functions)**.** *Given a family of selection functions $\varepsilon_i\colon (X_i \to R) \to X_i$, for $1 \leq i \leq n$, define their product by simply iterating the binary product as*

$$\bigotimes_{i=k}^{n} \varepsilon_i = \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{n} \varepsilon_i \right)$$

where $(\bigotimes_{i=n}^{n} \varepsilon_i) = \varepsilon_n$.

The interesting fact is that when the binary product of selection functions is iterated n times as above, it also calculates the backward induction reasoning for games with n rounds, as described in the following theorem. We omit the proof here, but give the full proof in the more general case of unbounded games in Section 3 (Theorem 4.6).

Theorem 2.8 (Backward Induction). *Given an n-player game $((X_i)_{i\in\mathbf{n}}, q)$ define a strategy profile as*

$$\mathsf{s}_i(x_1, \ldots, x_{i-1}) = \pi_0\left(\left(\bigotimes_{k=i+1}^{n} \mathsf{argsup}_k\right)(q_{x_0,\ldots,x_{i-1}})\right)$$

where π_0 denotes the first projection, i.e. $\pi_0(x * \alpha) = x$. Such strategy profile is in subgame perfect equilibrium.

2.3 Illustrative Example: Sweet Tooth Game

In the Sweet Tooth game [21] Jeremy and Marie have in front of them two identical rectangular cakes. Jeremy will cut the first cake into two pieces, in any way he chooses. Marie will look at the division of the first cake, and will decide whether she will chose first or allow Jeremy to do so. If she goes first, she will take the larger piece. If she goes second, she can assume that Jeremy will take the larger piece. Next, Jeremy will cut the second cake into two pieces. If Marie had chosen first, for the first cake, then Jeremy gets to take the larger piece of the second cake. If Marie had chosen second for the first cake, then she gets to take the larger piece of the second cake. Our task is to calculate a subgame perfect equilibrium strategy profile for Jeremy and Mary.

The game above has three rounds, with sets of moves

- $X_0 = [0, 1/2]$, the size of the smallest piece after Jeremy cut the first cake.
- $X_1 = \mathbb{B}$, the choice of Marie to either pick the largest piece or to leave it to Jeremy. Assume true means that Marie will pick the largest piece.
- $X_2 = [0, 1/2]$, the size of the smallest piece of the second cake, after Jeremy's cut.

The outcome function for the game is as follows

$$q(f, b, g) = \begin{cases} (f + 1 - g, 1 - f + g) & \text{if } b = \mathsf{true} \\ (1 - f + g, f + 1 - g) & \text{if } b = \mathsf{false}, \end{cases}$$

where $f, g \colon [0, 1/2]$ and $b \colon \mathbb{B}$, and the two components of $q(f, b, g)$ are the amount of cake Jeremy and Marie, respectively, get at the end of the game. In words, if Marie decides to choose on the first cake, she gets the largest piece of the first cake $1 - f$, and the smallest piece of the second g, leaving Jeremy with the smallest piece of the first f plus the largest of the second $1 - g$. If, on the other hand, she lets Jeremy have the largest piece of the first cake, she will get f of the first cake, and $1 - g$ of the second. The calculation of optimal strategies in this case via Theorem 4.6 involves the product of three selection functions (properly composed with the projections $\pi_0 \colon \mathbb{R}^2 \to \mathbb{R}$ and $\pi_1 \colon \mathbb{R}^2 \to \mathbb{R}$, respectively): $\mathsf{argsup}^{[0,1/2]} \colon ([0, 1/2] \to \mathbb{R}^2) \to [0, 1/2]$ on the first and third rounds, and $\mathsf{argsup}^{\mathbb{B}} \colon (\mathbb{B} \to \mathbb{R}^2) \to \mathbb{B}$ on the second round. The strategic play can be calculated as

$$(x, y, z) = (\mathsf{argsup}^{[0,1/2]} \otimes \mathsf{argsup}^{\mathbb{B}} \otimes \mathsf{argsup}^{[0,1/2]})(q).$$

According to Theorem 2.8, the optimal strategy for the last round is

$$\begin{aligned}
\mathsf{s}_2(f,b) &= \pi_0\left((\mathsf{argsup}^{[0,1/2]})(q_{f,b})\right) \\
&= \mathsf{argsup}^{[0,1/2]}(\lambda g.\left\{\begin{array}{ll} f+1-g & \text{if } b = \mathsf{true} \\ 1-f+g & \text{if } b = \mathsf{false}, \end{array}\right\}) \\
&= \left\{\begin{array}{ll} 0 & \text{if } b = \mathsf{true} \\ 1/2 & \text{if } b = \mathsf{false}. \end{array}\right.
\end{aligned}$$

This means, in the last round Jeremy will cut a vanishingly small part so he keeps the whole second cake, in case Marie chose to have the largest piece of the first cake. We then calculate the optimal strategy for Marie on the second round as follows, making use of the fact that we already know how Jeremy will play on round 2:

$$\begin{aligned}
\mathsf{s}_1(f) &= \pi_0\left((\mathsf{argsup}^{\mathbb{B}} \otimes \mathsf{argsup}^{[0,1/2]})(q_f)\right) \\
&= \mathsf{argsup}^{\mathbb{B}}(\lambda b.\left\{\begin{array}{ll} 1-f+\mathsf{s}_2(f,b) & \text{if } b = \mathsf{true} \\ f+1-\mathsf{s}_2(f,b) & \text{if } b = \mathsf{false}, \end{array}\right\}) \\
&= \mathsf{argsup}^{\mathbb{B}}(\lambda b.\left\{\begin{array}{ll} 1-f & \text{if } b = \mathsf{true} \\ f+1/2 & \text{if } b = \mathsf{false}, \end{array}\right\}) \\
&= \left\{\begin{array}{ll} \mathsf{true} & \text{if } f \leq 1/4 \\ \mathsf{false} & \text{if } f > 1/4. \end{array}\right.
\end{aligned}$$

So, Marie's optimal strategy is to choose the largest piece of the first cake if Jeremy cuts a small piece which is smaller than 1/4 of the cake. Finally, making use of the two strategies we have just calculated for rounds 1 and 2 we can compute Jeremy's optimal strategy (move!) for the first round:

$$\begin{aligned}
\mathsf{s}_0 &= \pi_0\left((\mathsf{argsup}^{[0,1/2]} \otimes \mathsf{argsup}^{\mathbb{B}} \otimes \mathsf{argsup}^{[0,1/2]})(q)\right) \\
&= \mathsf{argsup}^{[0,1/2]}(\lambda f.\left\{\begin{array}{ll} f+1-\mathsf{s}_2(f,\mathsf{s}_1(f)) & \text{if } \mathsf{s}_1(f) = \mathsf{true} \\ 1-f+\mathsf{s}_2(f,\mathsf{s}_1(f)) & \text{if } \mathsf{s}_1(f) = \mathsf{false}, \end{array}\right\}) \\
&= \mathsf{argsup}^{[0,1/2]}(\lambda f.\left\{\begin{array}{ll} f+1-\mathsf{s}_2(f,\mathsf{true})) & \text{if } f \leq 1/4 \\ 1-f+\mathsf{s}_2(f,\mathsf{false}) & \text{if } f > 1/4, \end{array}\right\}) \\
&= \mathsf{argsup}^{[0,1/2]}(\lambda f.\left\{\begin{array}{ll} f+1 & \text{if } f \leq 1/4 \\ 1-f+1/2 & \text{if } f > 1/4, \end{array}\right\}) \\
&= 1/4.
\end{aligned}$$

So, in an optimal play Jeremy will cut the first cake into the pieces 1/4 and 3/4, Marie will collect the bigger piece (3/4), he will then cut a vanishingly small bit of the second cake and take (essentially) the whole second cake, having an accumulated total cake amount of 1 and 1/4. In this case $(\mathsf{s}_0, \mathsf{s}_1, \mathsf{s}_2)$ is not only a strategy profile in subgame optimal equilibrium, but it is in fact the maximum amount of cake Jeremy can guarantee to have independently of how Marie plays (cf. [21]).

3 Unbounded Games

In this section we show how the binary product of selection function can be iterated an unbounded number of times, leading to a calculation of optimal strategies in subgame perfect equilibrium for unbounded games. Note that from now on we replace the indexing set $\mathbf{n} = \{1, 2, \ldots, n\}$ by the full set of natural number $\mathbb{N}$. Let $[\alpha](n)$ denote the first n elements of the infinite sequence $\alpha \colon \Pi_{i \in \mathbb{N}} X_i$. Let X^* denote the set of all finite sequence of the form $\Pi_{i<k} X_i$. We use $\mathbf{1}$ for the singleton type, and $X + Y$ for the disjoint union.

Definition 3.1 (Unbounded sequential game)**.** *An* unbounded sequential game *is given by a tuple* $(\{X_i\}_{i \in \mathbb{N}}, q)$ *where*

- X_i *is the* set of moves *for player* i*, and*
- $q \colon X^* \to (\mathbf{1} + \mathbb{R}^{\mathbb{N}})$ *is the* outcome function.

Note that we have potentially infinitely many players and hence potentially infinitely long plays. However, the outcome function $q \colon X^* \to (\mathbf{1} + \mathbb{R}^{\mathbb{N}})$ comes equipped with a flag which tells us when the game needs to be continued (i.e. has not terminated). If $q(s) \in \mathbf{1}$ the game must go on, whereas if $q(s) \in \mathbb{R}^{\mathbb{N}}$ the game has ended with payoffs $q(s)$. In order to ensure that our games are well-founded, we will require that any infinite play has a prefix which is a final position. Formally,

$$\forall \alpha \exists n \left(q([\alpha](n)) \in \mathbb{R}^{\mathbb{N}} \right). \tag{1}$$

Whenever this is assumed, we write $\omega(\alpha)$ for the least n satisfying (1). Under this assumption, the game tree is well-founded, although branches might be arbitrarily long as X_i are potentially infinite sets. We show how one can extend the formal description of backward induction given in the previous section to unbounded but well-founded games as described in Definition 3.1.

As in Section 2, we write $q_i \colon X^* \to (\mathbf{1} + \mathbb{R})$ for the composition of the outcome function q with the i-th projection $\pi_i \colon \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$, whenever $q(s) \in \mathbb{R}^{\mathbb{N}}$. Hence, given a play $\alpha \colon \Pi_{i \in \mathbb{N}} X_i$, the payoff of player i is given by $q_i(\alpha)$. We also define:

- A strategy for player i is a mapping $\mathsf{s}_i \colon \Pi_{k<i} X_k \to X_i$.
- A strategy profile is an infinite sequence of strategies $(\mathsf{s}_i)_{i \in \mathbb{N}}$ for each player.
- An strategy profile $(\mathsf{s}_i)_{i \in \mathbb{N}}$ determines a play α as

  $$s_i = \mathsf{s}_i([\alpha](i))$$

  and an outcome $u = q(s)$, where $|s|$ is the first point where $q(s) \in \mathbb{R}^{\mathbb{N}}$. We call this the strategic play (for the given strategy profile) and denote it by ν. For the strategic continuation of a partial play s we write $\nu(s)$.

Because the game can last for arbitrarily long, note that a strategy profile must include an infinite sequence of strategies, as we must be ready to play for longer and longer games depending on how the game unfolds.

Definition 3.2 (Unbounded Nash Equilibrium)**.** *Let us fix an unbounded game* $(\{X_i\}_{i \in \mathbb{N}}, q)$ *where* q *satisfies the well-foundedness condition (1). Given a strategy profile* $(\mathsf{s}_i)_{i \in \mathbb{N}}$ *let* s *denote the (finite!) play it determines. Such strategy profile is said to be in* unbounded Nash equilibrium *if for each player* $i \leq |s|$ *and alternative move* $x \in X_i$ *at point* i

$$q_i(\nu) \geq q_i([\nu](i) * x * \nu([\nu](i) * x)).$$

The formula $[\nu](i) * x * \nu([\nu](i) * x)$ describes the play which starts by following the strategy $(\mathsf{s}_i)_{i\in\mathbb{N}}$, then at round i one plays an arbitrary move x, and then continues to follow the strategy from the partial play $[\nu](i) * x$ onwards.

Note that we only require the equilibrium to happen up to the point where the game finishes on the play s determined by the strategy profile $(\mathsf{s}_i)_{i\in\mathbb{N}}$. However, it is crucial to observe that such point might be arbitrarily long depending on how the game unfolds. As in Definition 2.4, we can similarly define a subgame of an unbounded game.

Definition 3.3 (Unbounded Subgame Perfect Equilibrium)**.** *A strategy profile $(\mathsf{s}_i)_{i\in\mathbb{N}}$ is in* unbounded subgame perfect equilibrium *is it is in unbounded Nash equilibrium on any subgame.*

4 Backward Induction for Unbounded Games

The binary product of selection function (Definition 2.6) can also be iterated an unbounded number of times as follows:

Definition 4.1 (Unbounded Product)**.** *Given an outcome function $q\colon X^* \to (\mathbf{1} + R)$ and an infinite sequence of selection functions $\varepsilon_i\colon (X_i \to (\mathbf{1}+R)) \to X_i$, define the* unbounded product *of $(\varepsilon_i)_{i\in\mathbb{N}}$, denoted by $\mathsf{BI}_i(\varepsilon)$, by simply iterating the binary product of selection functions as*

$$\mathsf{BI}_i(\varepsilon)(q) = \begin{cases} \langle\,\rangle & \textit{if} \quad q(\langle\,\rangle) \in R \\ (\varepsilon_i \otimes \mathsf{BI}_{i+1}(\varepsilon))\,(q) & \textit{if} \quad q(\langle\,\rangle) \in \mathbf{1}. \end{cases}$$

Note that $\mathsf{BI}_i(\varepsilon)$ itself is a selection function of type $(X^* \to (\mathbf{1} + R)) \to X^*$, and that is why the binary product can be used in the definition of $\mathsf{BI}_i(\varepsilon)$ above. It is easy to see, however, that by expanding the definition of the binary product $\otimes$ (Definition 2.6) the functional $\mathsf{BI}_i(\varepsilon)$ can be equivalently defined as

$$\mathsf{BI}_i(\varepsilon)(q) = \begin{cases} \langle\,\rangle & \text{if} \quad q(\langle\,\rangle) \in R \\ a_i * \mathsf{BI}_{i+1}(\varepsilon)(q_{a_i}) & \text{if} \quad q(\langle\,\rangle) \in \mathbf{1}, \end{cases} \tag{2}$$

where $a_i = \varepsilon_i(\lambda x.q_x(\mathsf{BI}_{i+1}(\varepsilon)(q_x)))$.

Lemma 4.2. *Assuming the outcome function $q\colon X^* \to (\mathbf{1} + R)$ satisfies (1) the functional $\mathsf{BI}_i(\varepsilon)$ is well-defined.*

Proof. Fix q satisfying (1), and assume for some i and ε the value of $\mathsf{BI}_i(\varepsilon)(q)$ is undetermined. That can only be because $q(\langle\,\rangle) \in \mathbf{1}$ and $\mathsf{BI}_{i+1}(\varepsilon)(q_{a_i})$ is also undetermined. Again, $\mathsf{BI}_{i+1}(\varepsilon)(q_{a_i})$ can only be undetermined if $q_{a_i}(\langle\,\rangle) = q(\langle a_i\rangle) \in \mathbf{1}$ and $\mathsf{BI}_{i+2}(\varepsilon)(q_{a_i,a_{i+1}})$ is also undetermined. Continuing like this we find an infinite sequence of values $\alpha = a_i, a_{i+1}, \ldots$ such that for all k we have $q([\alpha](k))) \in \mathbf{1}$, as $[\alpha](k) = a_i, \ldots, a_{i+k}$, which contradicts (1). $\square$

We now prove three important properties of the functional $\mathsf{BI}_i(\varepsilon)$ which are going to lead us to the main theorem below. The first of these characterises the k-th component of the sequence $\mathsf{BI}_i(\varepsilon)(q)\colon X^*$.

Lemma 4.3. *Let $t = \mathsf{BI}_i(\varepsilon)(q)$. Then, for $0 \le k < |t|$,*

$$t_k = \varepsilon_k(\lambda x.q_{[t](k)*x}(\mathsf{BI}_{i+k+1}(\varepsilon)(q_{[t](k)*x}))).$$

Proof. By a simple induction on k. □

It is interesting to think of sequences $s = x_0, \ldots, x_{i-1}$ as partial plays in the game. The next lemma says that given a partial play s, the continuation of such play calculated by $\mathsf{BI}_{|s|}(\varepsilon)(q_s)$ is in some sense "idempotent", so that recalculating the continuation of the play after a few steps results in the same play originally computed. This is made precise as follows:

Lemma 4.4. *Fix s and let*

$$t = \mathsf{BI}_{|s|}(\varepsilon)(q_s).$$

We think of q_s as the subgame determined by the partial play s, and t as the continuation of the partial play s according to the product of selection functions. For all $0 \leq i < |t|$ the following holds:

$$t = [t](i) * \mathsf{BI}_{|s|+i}(\varepsilon)(q_{s*[t](i)}).$$

Proof. By a simple induction on i. If $i = 0$ this follows by the definition of t. Assume this holds for i we wish to show it for $i+1$. We have

$$t \overset{\text{(IH)}}{=} [t](i) * \mathsf{BI}_{|s|+i}(\varepsilon)(q_{s*[t](i)}) \overset{(2)}{=} [t](i) * a_i * \mathsf{BI}_{|s|+i+1}(\varepsilon)(q_{s*[t](i)*a_i}),$$

where $a_i = \varepsilon_{|s|+i}\left(\lambda x.q_{s*[t](i)*x}(\mathsf{BI}_{|s|+i+1}(\varepsilon)(q_{s*[t](i)*x}))\right)$. Hence, $t_i = a_i$. Therefore, we have that $t = [t](i+1) * \mathsf{BI}_{|s|+i+1}(\varepsilon)(q_{s*[t](i+1)})$. □

Let us denote by $\mathsf{argsup}_i \colon (X_i \to (\mathbf{1} + \mathbb{R}^{\mathbb{N}})) \to X_i$ the composition of the i-th projection $\pi_i \colon (\mathbf{1} + \mathbb{R}^{\mathbb{N}}) \to \mathbb{R}$ with $\mathsf{argsup} \colon (X_i \to \mathbb{R}) \to X_i$. For definiteness we say that $\pi_i(1) = 0$. For the rest of this section we fix an unbounded sequential game $((X_i)_{i\in\mathbb{N}}, q)$, and the following strategy profile

$$\mathsf{s}_i(t) = \pi_0\left(\mathsf{BI}_{|t|}(\mathsf{argsup})(q_t)\right). \tag{3}$$

Our aim is to show that such strategy profile is in unbounded subgame perfect equilibrium.

Lemma 4.5. *Given a partial play s its strategic extension can be calculated by $\mathsf{BI}_{|s|}(\mathsf{argsup})(q_s)$, i.e.*

$$\nu(s) = \mathsf{BI}_{|s|}(\mathsf{argsup})(q_s).$$

Proof. Let $t = \mathsf{BI}_{|s|}(\mathsf{argsup})(q_s)$. We have to show that

$$t_i = \mathsf{s}_i(s * [t](i)) \overset{(3)}{=} \pi_0\left(\mathsf{BI}_{|s|+i}(\mathsf{argsup})(q_{s*[t](i)})\right).$$

This follows directly from Lemma 4.3, since

$$\begin{aligned} \pi_0\left(\mathsf{BI}_{|s|+i}(\mathsf{argsup})(q_{s*[t](i)})\right) &\overset{(2)}{=} \mathsf{argsup}_{|s|+i}(\lambda x.q_{s*[t](i)*x}(\mathsf{BI}_{|s|+i+1}(\mathsf{argsup})(q_{s*[t](i)*x}))), \\ &\overset{\text{L. 4.3}}{=} \mathsf{BI}_{|s|}(\mathsf{argsup})(q_s)(i) = t_i, \end{aligned}$$

which concludes the proof. □

Finally, we arrive at our main result:

Theorem 4.6 (Backward Induction for Unbounded Games)**.** *The strategy profile defined in (3) is in subgame perfect equilibrium.*

Proof. Let s be a arbitrary partial non-terminating play. Let t be its strategic extension. By the definition of the strategic move at round k we have

$$\begin{aligned} t_i &= \mathsf{s}_{|s|+i}(s * [t](i)) \\ &\overset{(3)}{=} \pi_0\left(\mathsf{BI}_{|s|+i}(\mathsf{argsup})(q_{s*[t](i)})\right). \\ &\overset{(2)}{=} \mathsf{argsup}_{|s|+i}\left(\lambda x. q_{s*[t](i)*x}\left(\mathsf{BI}_{|s|+i+1}(\mathsf{argsup})(q_{s*[t](i)*x})\right)\right). \end{aligned}$$

By the definition of argsup we have

$$q_i(s*[t](i)*t_i*\left(\mathsf{BI}_{|s|+i+1}(\mathsf{argsup})(q_{s*[t](i)*t_i})\right)) \geq q_i(s*[t](i)*x*\left(\mathsf{BI}_{|s|+i+1}(\mathsf{argsup})(q_{s*[t](i)*x})\right))$$

for any x. By Lemma 4.5 we have that $\mathsf{BI}_{|s|+i+1}(\mathsf{argsup})(q_{s*[t](i)*x}) = \nu(s*[t](i)*x)$, so that the above simplifies to

$$q(s*[t](i+1)*\nu(s*[t](i+1))) \geq q(s*[t](i)*x*\nu(s*[t](i)*x)).$$

Finally, by Lemma 4.4 we have $s*[t](i+1)*\nu(s*[t](i+1)) = \nu(s)$, and the result follows. □

4.1 Generalizations

The functional $\mathsf{BI}_i(\varepsilon)$ described in (2) and used to calculate optimal strategies in the previous section is a form of recursion on well-founded trees known as *bar recursion*. It was first investigated in the context of proof theory by Spector [22] where he extends Gödel's relative consistency proof of Peano arithmetic [15] to full analysis. Several other variants of bar recursion have been considered recently (see e.g. [5, 6, 16]) and their inter-definability has been investigated by the authors in [8].

We note here that the more powerful forms of bar recursion considered by Spector [22] and Berardi, Bezem and Coquand [5] (called EPS and IPS, respectively, in [8]) rather than $\mathsf{BI}_i(\varepsilon)$ above would allow us to compute subgame optimal equilibria for other variants of games of unbounded length. In the case of EPS, for instance, we could deal with games where the outcome function $q\colon \Pi_i X_i \to R$ does not explicitly inform us about the termination of the game, but an extra auxiliary function $\omega\colon \Pi_i X_i \to \mathbb{N}$ tells us about what "relevant" part of any infinite play actually is. That allows us to compute an equilibrium profile which is optimal in the relevant part of the strategic play. This is discussed in detail in [11] for a general notion of sequential games introduced in [9].

5 Conclusions

In [9, 10, 11] we worked with essentially the same formalization of sequential games, and showed that the product of selection functions calculates *optimal strategies*. What is new in this paper is that, for the particular case of sequential game where the set of outcomes is $\mathbb{R}^n$ and the selection functions are argsup_i, the product of selection functions turns out to compute *subgame perfect Nash equilibria*. That is, for such games the notion of optimal strategy coincides with the notion of subgame perfect equilibrium (a refinement of Nash equilibrium), and both are given by the product of selection functions. This gives an explicit formula for their computation, which can be directly interpreted as an algorithm in Gödel's system T (extended with the unbounded product), or in any practical functional programming language (without any extensions) [10].

A large amount of literature on backward induction and subgame perfect equilibrium discusses cases where this notion of equilibrium leads to strategies which are, from a common sense point of view, not optimal. For instance, the Centipede game of Rosenthal [19] is a standard example which illustrates some unexpected outcome of the backward indunction method, whereby the subgame perfect equilibrium leads to a considerably smaller payoff than what could have been obtained by either player collectively changing their strategies. It is in fact widely known that strategy profiles which are in subgame perfect equilibrium are often not what one in practice (experimentally) will think of as the best strategy. That is mainly because common knowledge of rationality or common belief play an important role in sequential games. This led people to investigate the relation between common knowledge and backward induction, for instance Aumann [3] who shows that common knowledge rationality implies backward induction, or Ben-Porath [4] who investigates the characterization of the set of outcomes that are consistent with common certainty of rationality at the beginning of a game.

References

[1] C. D. Aliprantis. On the backward induction method. *Economic Letters*, 64:125–131, 2001.

[2] Geir B. Asheim. Backward induction. In W. Leinfellner, G. Eberlein, and S. H. Tijs, editors, *The Consistent Preferences Approach to Deductive Reasoning in Games*, volume 37 of *Theory and Decision Library*, pages 79–97. Springer US, 2006.

[3] Robert J. Aumann. Backward induction and common knowledge of rationality. *Games and Economic Behavior*, 8:6–19, 1995.

[4] E. Ben-Porath. Rationality, Nash equilibrium, and backwards induction in perfect information games. *Review of Economic Studies*, 64:23–46, 1997.

[5] S. Berardi, M. Bezem, and T. Coquand. On the computational content of the axiom of choice. *The Journal of Symbolic Logic*, 63(2):600–622, 1998.

[6] U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16:163–183, 2006.

[7] Luc Bovens. The backward induction argument for the finite iterated prisoner's dilemma and the surprise exam paradox. *Analysis*, 57(3):179–186, 1997.

[8] M. H. Escardó and P. Oliva. Computational interpretations of analysis via products of selection functions. In F. Ferreira, B. Lowe, E. Mayordomo, and L. M. Gomes, editors, *Computability in Europe 2010, LNCS*, pages 141–150. Springer, 2010.

[9] M. H. Escardó and P. Oliva. Selection functions, bar recursion, and backward induction. *Mathematical Structures in Computer Science*, 20(2):127–168, 2010.

[10] M. H. Escardó and P. Oliva. What sequential games, the Tychnoff theorem and the double-negation shift have in common. *To appear: MSFP 2010 (ACM SIGPLAN Mathematically Structured Functional Programming)*, 2010.

[11] M. H. Escardó and P. Oliva. Sequential games and optimal strategies. *Royal Society Proceedings A*, 467:1519–1545, 2011.

[12] Christian Ewerhart. Backward induction and the game-theoretic analysis of chess. *Games and Economic Behavior*, 39:206–214, 2002.

[13] J. Gale, K. G. Binmore, and L. Samuelson. Learning to be imperfect: The ultimatum game. *Games and Economic Behavior*, 8:56–90, 1995.

[14] Herbert Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic*. Princeton University Press, 2009.

[15] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.

[16] U. Kohlenbach. *Theorie der majorisierbaren und stetigen Funktionale und ihre Anwendung bei der Extraktion von Schranken aus inkonstruktiven Beweisen: Effektive Eindeutigkeitsmodule bei besten Approximationen aus ineffektiven Eindeutigkeitsbeweisen.* PhD thesis, Frankfurt, pp. xxii+278, 1990.

[17] M. J. Osborne. *An Introduction to Game Theory.* Oxford University Press, USA, 2004.

[18] Andrés Perea. *Rationality in Extensive Form Games.* Game Theory, Mathematical Programming and Operations Research. Kluwer Academic Publishers, Boston, 2001.

[19] R. Rosenthal. Games of perfect information, predatory pricing, and the chain store. *Journal of Economic Theory*, 25:92–100, 1981.

[20] U. Schwalbe and P. Walker. Zermelo and the early history of game theory. *Games and Economic Behavior*, 34:123–137, 2001.

[21] Dennis Shasha. *Puzzles for Programmers and Pros.* John Wiley & Sons, 2007.

[22] C. Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In F. D. E. Dekker, editor, *Recursive Function Theory: Proc. Symposia in Pure Mathematics*, volume 5, pages 1–27. American Mathematical Society, Providence, Rhode Island, 1962.

[23] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, New Jersey, 1944.

[24] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[25] E. Zermelo. Ueber eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proceedings of the Fifth International Congress of Mathematicians*, volume II, pages 501–504. Cambridge University Press, Cambridge, 1912.